(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 January 2003 (09.01.2003)

PCT

(10) International Publication Number
WO 03/003695 A1

(51) International Patent Classification[7]: H04L 29/06

(21) International Application Number: PCT/US02/20545

(22) International Filing Date: 28 June 2002 (28.06.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/302,455	29 June 2001 (29.06.2001)	US
10/184,286	27 June 2002 (27.06.2002)	US

(71) Applicant: **MOTOROLA, INC.** [US/US]; 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

(72) Inventors: **O'HARE, John, B.**; Abbeydale, Temple Hill Lawn, Ballintemple, Cork (IE). **ALLEN, Andrew, M.**; 1937 McRae Lane, Mundelein, IL 60060 (US). **MAGEE, Stephen, D.**; 12521 N. 76th Street, Scottsdale, AZ 85260 (US). **CHEUNG, Collen, Yue, Ling**; 19 Wakefield Close, Swindon, Wiltshire SN5 8QU (GB). **MAJUMDAR, Ishita**; 1204 E. Algonquin Road, #2N, Schaumburg, IL 60173 (US). **FOSTER, Gerry, T.**; 23 Reynolds Way, St. Andrew's Ridge, Swindon, Wiltshire SN25 4GF (GB).

(74) Agents: **WILLIAMS, Lalita, P.** et al.; Motorola, Inc., 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

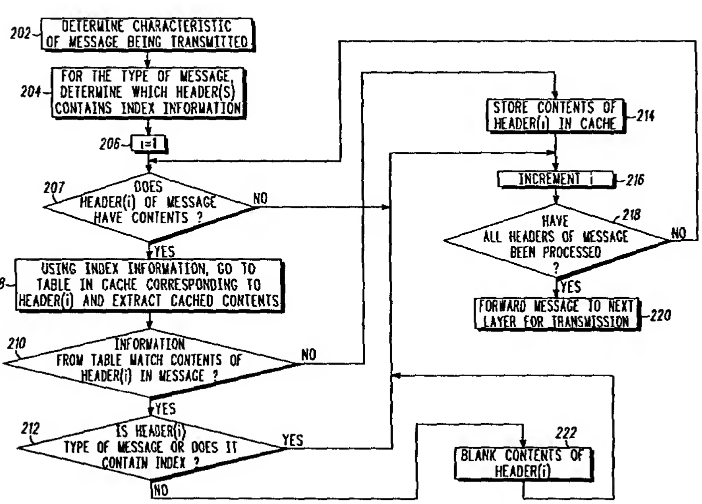
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** APPARATUS AND METHOD FOR IMPLEMENTING TEXT BASED COMPRESSION USING CACHE AND BLANK APPROACH



(57) Abstract: An apparatus and method for compressing and compressing Text based messages. A TCCB layer 114, 115 is added to a mobile device 100 and P-CSCF 102. At the sending device, the TCCB compression method is invoked to remove all redundant header information contained in a message to be transmitted. When the TCCB compression method detects that a header's contents are the same as previously sent or received for a particular sequence, the TCCB method blanks the header contents. At the receiving device, when the TCCB layer receives a message with blank header contents, the TCCB decompression method is invoked to reconstruct the header from its cache 115, 127.

WO 03/003695 A1

APPARATUS AND METHOD FOR IMPLEMENTING TEXT BASED COMPRESSION USING CACHE AND BLANK APPROACH

Field of the Invention

The present invention relates generally to the field of communication systems, and more particularly, to text based compression schemes.

Background of the Invention

Currently, telephony service is provided for the most part over circuit switched networks. A fast emerging new trend called IP telephony provides telephony service over Internet Protocol (IP) networks. The motivating factors for carrying voice traffic over data networks are the integration of voice and data applications, which can result in more effective business process, cost savings for voice calls and enabling of many new services for business and customers. The flexibility offered by IP telephony lies in moving the intelligence from the network to the end stations, thereby enabling many new services that did not exist before. In an effort to merge Internet and cellular telephony, two aspects are focused on -- end-to-end call set up delay and voice quality.

Protocols such as Session Initiated Protocol (SIP) and Session Description Protocol (SDP) will typically be used to set up and tear down calls. However, adopting ASCII based protocols such as SIP and SDP in access networks of limited bandwidth incurs a significant delay for call set up. Passing large text messages over the air interface also results in a very inefficient use of the transmission medium. In addition, some legacy based enhanced time division multiplexed (TDM) cellular transceivers, such as GSM EDGE Radio Access Network (GERAN), will need to "steal" audio bandwidth in order to transmit in-

call SIP signaling messages. This stealing of audio bandwidth will likely result in long audio mutes.

Thus there is a need for a method of compressing text based messages in order to increase spectrum efficiency, reduce transmission delay and provide a comparable level of quality of service compared with circuit switched systems.

Brief Description of the Drawings

FIG. 1 is a block diagram of an architecture that can be used to implement the TCCB apparatus and method of the present invention.

FIG. 2 is a flow diagram of the preferred embodiment of the TCCB compression method of the present invention.

FIG. 3 is a flow diagram of the preferred embodiment of the TCCB decompression method of the present invention.

FIG. 4 is a snapshot of the mobile device cache after sending a registration message for the first time.

FIG. 5 is a snapshot of the core network cache after a registration message is processed by the TCCB layer for the first time.

FIG. 6 is a snapshot of the mobile device cache after an "INVITE" message is processed by the TCCB layer for the first time.

Detailed Description of the Drawings

The present invention provides a method and apparatus for compressing text based messages in order to increase spectrum efficiency, reduce transmission delay and provide a comparable level of quality of service compared with circuit switched systems. In particular, the present invention provides an apparatus and method for performing Text-based Compression using a Cache and Blank approach (TCCB). TCCB is designed to be extensible. It can work over various access technologies and the principle applies to any text based protocols. An advantage of the TCCB method is that only the

User Equipment (UE) (e.g. mobile device) and the Peer Core Network Entity (e.g. Proxy CSCF) need be involved in the storage and retrieval of information to compress or decompress the text based messages.

FIG. 1 is a block diagram of an architecture that can be used to implement theTCCB apparatus and method of the present invention. In the preferred embodiment, the architecture includes a mobile device 100 and a Proxy CSCF (P-CSCF) 102 coupled to a Radio Access Network (RAN) 104. The mobile device 100 is coupled to the RAN 104 through an air interface 106, while the P-CSCF 102 is coupled to the RAN 104 through an IP interface 108. Refer to Section 5 of TR 23.821 v1.0.0 (2000-06) and Section 5.6.1.1 of 3GPP TS 23.060 v4.0.0 (2001-03) for a more detailed description of the functionality and connectivity of the aforementioned components. As known in the art, the software architecture of the mobile device 100 and P-CSCF 102 typically includes a seven (7) layer stack running on a microprocessor. The stack includes from top to bottom, an application layer 110, 124; a Text Based Protocol (TBP) layer 112, 126; a User Datagram Protocol (UDP) layer 116, 130; a Transmission Control Protocol (TCP) layer 118, 132; an IP layer 120, 134; and a physical layer 122, 136. The UDP, IP and TCP layers are generally referred to as the lower transport layers. A detailed description of the OSI reference model (7 layered model) can be found in "Computer Networks" (2nd Edition) by Andrew S. Tannenbaum, pgs. 15-20. The present invention introduces an additional TCCB layer 114, 128. In the preferred embodiment, the TCCB layer 114, 128 is positioned between the TBP layer 112, 126 and the lower transport layers. The TCCB layer 114, 128 is added so that the mobile device 100 and P-CSCF 102 can communicate using compressed messages. In the mobile device 100, the TCCB layer 114 includes a mobile device cache (MDC) 115. In the P-CSCF 102, the TCCB layer 128 includes a core network cache (CNC) 127. As will be described with reference to the TCCB method of the present invention,

the MDC 115 and CNC 127 are used to store tables of information used to compress and decompress messages transmitted between the mobile device 100 and the P-CSCF 102. The TCCB method of the present invention can be implemented using any commercially available microprocessors.

The TCCB layer 114, 128 in both the mobile device 100 and the P-CSCF 102 includes the compression method and decompression method of the present invention. When either the mobile device 100 or P-CSCF 102 prepares to transmit a message, the TCCB compression method is invoked to remove all redundant header information contained in the message. Redundant header information is information originating from the sending device that is already stored in the MDC 115 or CNC 127 of the receiving device. This could occur, for example, when a header's contents have not changed from a previously transmitted message for a particular session. When the TCCB compression method detects that a header's contents are the same as previously sent or received for a particular sequence, the TCCB method blanks the header contents. At the receiving device, when the TCCB layer receives a message with blank header contents, the TCCB decompression method is invoked to reconstruct the header from its cache. Details of the compression method and decompression method will be described with reference to the flow charts of FIGs. 2 and 3.

FIG. 2 is a flow diagram of the preferred embodiment of the TCCB compression method of the present invention. At step 202, the method determines the characteristic (e.g. type) of the message being transmitted. For the type of message, the method determines which header(s) of the message contains the correct index information into the cache 115 or 127 (step 204). For example, if the mobile device 100 is transmitting the message, the correct index information into cache 115 is determined at step 204. At step 206, the method

initializes a variable “i.” This variable is used to track the particular header of the message that is being operated on. At step 207, the method determines whether the header(i) has contents. If the header has contents,at step 208, the method uses the index information into the MDC 115 or CNC 127 and extracts the contents of header(i) from a table. At step 210, the method determines whether the information from the table matches the contents of header(i) in the message. If the information does not match, the method stores the contents of header(i) from the message in the table at step 214. At step 216, the method increments “i” to point to the next header for processing. At step 218, the method determines whether all headers of the message have been processed. If all headers have been processed, the method forwards the message to one of the lower transport layers 116, 118, 120, 130, 132, 134 (FIG. 1) for transmission (step 220). If all headers have not been processed, the method returns to step 208 to continue processing the remaining headers.

Referring back to step 207, if the header(i) does not have contents, the method proceeds to step 216 where “i” is incremented to continue processing of any remaining headers. Referring back to step 210, if the information from the table matches the contents of header(i) in the message, at step 212 the method determines whether header(i) is the type of message or whether header (i) contains the index information into the cache. If header(i) is not the type of message and does not contain the index information, the method blanks the contents of header(i) (step 222). Next, the method increments “i” (step 216) and proceeds to continue processing of any remaining headers. If at step 212 the method determines that the header(i) is the type of message or contains the index information into the cache, the method skips step 222 and increments “i” at step 216 to continue processing of any remaining headers. In the preferred embodiment, the header

representing the type of message (Request Line) and the header containing the index information can not be blanked.

FIG. 3 is flow diagram of the preferred embodiment of the TCCB decompression method of the present invention. At step 302, the method determines the characteristic (e.g. type) of the message being received. For the characteristic determined, the method determines which header(s) of the message contains the correct index information into the cache 115 or 127 (step 304). For example, if the P-CSCF 102 is receiving the message, the correct index information into cache 127 will be determined at step 304. At step 306, the method initializes variable "i" to 1. At step 307, the method determines whether header(i) includes a message name and protocol. If the answer is no, at step 308, the method determines whether the contents of header(i) are blank. If the contents are not blank, the method uses the index information into the MDC 115 or CNC 127 and stores the contents of header(i) in the received message in the table corresponding to header(i) (step 312). At step 314, the method increments "i." At step 316, the method determines whether all headers of the message have been processed. If all headers have been processed, at step 318, the method forwards the message to the TBP layer 112, 126 (FIG. 1) for processing. If all headers have not been processed, the method repeats step 308 to continue processing the remaining headers.

Referring back to step 307, if the answer is yes, the method proceeds to step 314 where "i" is incremented to continue processing of any remaining headers. Referring back to step 308, if the contents of header(i) in the message are blank, the method uses the index information to access the cache table corresponding to header(i) and extracts the information from the table to fill in the contents of header(i) in the message (step 310). Next, the method increments "i" at step 314 to continue processing of any remaining headers.

To illustrate how the TCCB compression and decompression methods operate, an example registration of a mobile device 100 with a P-CSCF 102 will now be described. The example features SIP messages exchanged between the mobile device 100 and P-CSCF 102. However, any text-based messages can be used with the present invention. The first example is a first registration after power up of the mobile device 100. To initiate the process, a REGISTER request is sent from the application layer 110 in the MD 100 (FIG. 1) to the TCCB layer 114. In the current example, the message has the following fields (headers):

REGISTER sip:bell-tel.com SIP/2.0
Via: SIP/2.0/UDP saturn.bell-tel.com
From: <sip:watson@bell-tel.com>; tag=19a1
To: sip:watson@bell-tel.com
Call-ID: 70710@saturn.bell-tel.com
Cseq: 1 REGISTER
Contact: <sip:watson@saturn.bell-tel.com:3890;
transport=udp>
Expires: 7200

The message contains eight (8) headers: REGISTER, Via, From, To, Call-ID, Cseq, Contact and Expires. The content of each header is the information shown after the colon, with the exception of the first line (Request Line) where the content is everything after the message name. For example, the contents of the "REGISTER" header is "sip:bell-tel.com SIP/2.0." The contents of the "Via" header is "SIP/2.0/UDP saturn.bell-tel.com."

The TCCB layer 114, 128 invokes the compression method to compress the message (if possible) before sending it to the transport layers 116, 118, 120, 130, 132, 134 for transmission. Referring to the

compression algorithm of FIG. 2, the method determines at step 202 that the message is a “REGISTER” message. This determination is made by looking at the Request Line (first line) of the message. At step 204, the method determines which header contains the correct index information into a table in the MDC 115 corresponding to the Request Line. In the preferred embodiment, the “To” header contains the index information for a “REGISTER” message. In an alternate embodiment, a different header or combination of headers may contain the index information. At step 206, the method initializes variable “i” to 1 to point to the first header (header (1)) of the REGISTER message. At step 207, the method determines that the REGISTER message has contents and at step 208, uses the contents of the “To” header as index information into the Request Line table in the MDC 115. In the current example, “sip:watson@bell-tel.com” is the correct index information. At step 210, the method determines whether the contents of the Request Line table match the contents of header(1) which, in the current example, is “sip:bell-tel.com SIP/2.0.” Since, this is the first message transmitted, the message is not already stored in the MDC 115. At step 214, the method stores “sip:bell-tel.com SIP/2.0” in the Request Line table of the MDC 115.

At step 216, the method increments “i” to point to the “Via” header (header (2)) for processing. At step 218, the method determines that there are more headers to be processed and returns to step 207. The method repeats steps 207, 208, 210, 214, 216 and 218 until all headers have been processed. The method checks the contents of each header in the message against its corresponding table in the MDC 115 using the “To” header contents as the index information. Because this is the first message sent, none of the header contents are already stored in the MDC 115 and none of the header contents will be blanked. When the last header has been processed, the message is sent uncompressed (as shown above) to one of the

lower transport layers 114, 118, 120 for transmission to the RAN 104 over the air interface 106. The message is stored in the MDC 115 as shown in FIG. 4. The RAN 104 forwards the message to the P-CSCF 102 via the IP interface 108.

Upon receiving the message, the P-CSCF 102 invokes the decompression method of the present invention. Referring to FIG. 3, at step 302 the method determines that it received a REGISTER message. At step 304, the method determines that for a REGISTER message, the "To" header contains the index information into the CNC 127 (FIG. 1). At step 306, the method initializes "i" to 1 to point to the first header of the received message. At step 307, the method determines that header(1) does not include a message name and protocol and, at step 308, determines that the contents of header(1) are not blank. At step 312, the method uses the index information to update the Request Line table with the contents of header (1), "sip:bell-tel.com SIP/2.0." At step 314, the method increments "i" to point to header(2). At step 316, the method determines that there are more headers to be processed and repeats steps 307, 308, 312, 314 and 316 until all headers have been processed. Once all headers are processed, the message is forwarded to the TBP layer 126 for processing (step 318). The CNC 127 contains the values shown in FIG. 5.

In the current example, the P-CSCF 102 responds to the "REGISTER" message with a "200 OK" message. The 200 OK message is as follows:

SIP/2.0 200 OK
Via: SIP/2.0/UDP Saturn.bell-tel.com
From: <sip:Watson@bell-tel.com>; tag=19a1
To: sip:Watson@bell-tel.com
Call-ID: 70710@saturn.bell-tel.com

Cseq: 1 REGISTER

Contact: <sip:Watson@Saturn.bell-tel.com:3890; transport=udp>

It should be noted that the 200 0K message has no contents. The header of the 200 0K message is "SIP/2.0 200 0K." The P-CSCF 102 invokes the compression method to determine whether any of the header contents can be blanked before transmission to the mobile device 100. Referring to FIG. 2, the method determines that the message is a 200 OK message (step 202). At step 204, the method determines that the "To" header contains the index information for a 200 OK message. (In an alternate embodiment, another header or combination of headers may contain the index information.) At step 206, the method initializes variable "i" to 1 to point to the Request Line header (header (1)). At step 207, the method determines that header(1) has no contents and proceeds to step 216 to increment "i." At step 218, the method determines that there are more headers to be processed and proceeds to step 207. At step 207, the method determines that header(2), the "Via" header has contents. At step 208, the method uses "sip:watson@bell-tel.com" as the index information into the Via table of the CNC 127. As shown in FIG. 5, the contents of the Via table are "SIP/2.0/UDP saturn.bell-tel.com." Thus, at step 210, the method determines that the contents of the Via table in the CNC 127 match the contents of the Via header in the message (header(2)). At step 212, the method determines that header(2) is not the type of message and does not contain the index information and blanks the contents of the Via header (step 222). Next, the method proceeds to step 216 to increment "i".

At step 218, the method determines that there are more headers to be processed and proceeds to step 207. At step 207, the method determines that header(3) has contents and at step 208, uses

"sip:watson@bell-tel.com" as index information into the "From" table of the CNC 127. As shown in FIG. 5, the content of the "From" table is "<sip:watson@bell-tel.com>; tag=19a1" which matches the content of the From header in the 200 OK message. At step 210, the method determines there is a match and at step 212 determines whether the From header indicates the type of message or contains the index information. Since the answer to both questions is no, the method blanks the contents of the From header (step 222) and increments "i" at step 216 to process the "Call-ID" header (header(4)).

After the remaining headers are processed, the compressed 200 OK message is:

SIP/2.0 200 OK
Via:
From:
To: sip:watson@bell-tel.com
Call-ID:
Cseq:
Contact:

The compressed message is sent to one of the lower transport layers 130, 132, 134 for transmission to the mobile device 100.

Upon receipt of the compressed 200 OK message in the TCCB layer 114 of the mobile device 100, the decompression method is invoked. Referring to FIG. 3, the method determines that a "200 OK" message has been received (step 302). At step 304, the method determines that the contents of the "To" header contains the index information into the MDC 115. At step 306, variable "i" is initialized to 1. At step 307, the method determines that the header(1) includes a message name (200 OK) and protocol (SIP) and at step 314

increments “i.” At step 316, the method determines that there are more headers to be processed and proceeds to step 307. At step 307, the method determines that header(2) does not include a message name and protocol. At step 308, the method determines that the contents of header(2) are blank and uses “sip:watson@bell-tel.com” to extract the information from the “Via” table (SIP/2.0/UDP saturn.bell-tel.com) to fill in the contents of the “Via” header in the message. At step 314, the method increments “i” to point to header(3). At step 316, the method determines that there are more headers to be processed and proceeds to step 308. The method repeats steps 307, 308, 310, 312, 314 and 316 until all headers in the message have been filled in from information in the MDC tables. At step 318, the decompressed message will be forwarded to the TBP layer 112 for processing.

The compression and decompression methods of the present invention can also be implemented using multiple level indexing into the MDC 115 and CNC 127. As an example, consider an “INVITE” message transmitted from the mobile device 100 to the P-CSCF 102 after the mobile device 100 has successfully registered with the P-CSCF 102. For an INVITE message, the compression and decompression methods use both the “From” and “To” header contents as the index information into the MDC 115 and CNC 127, when necessary. The “From” contents provide the first level of indexing while the “To” offers a further level of granularity, should it be required. An example INVITE message with SDP information removed is:

INVITE sip:simpson@springfield.bell-tel.com SIP/2.0
Via: SIP/2.0/UDP saturn.bell-tel.com
From: T. Watson <sip:watson@bell-tel.com>
To: H. Simpson <sip:simpson@bell-tel.com>
Call-ID: 662606876@saturn.bell-tel.com
CSeq: 6 INVITE

Contact: <sip:watson@saturn.bell-tel.com>
Subject: Mr. Simpson, Going to Moe's?
Content-Type: application/sdp

When the TCCB layer 114 receives the INVITE message, it invokes the compression algorithm. Referring to FIG. 2, the method determines that the message is an INVITE message (step 202). At step 204, the method determines that the "From" and "To" header (when necessary) contains the index information for an INVITE message. At step 206, the method initializes variable "i" to 1 to point to the Request Line header (header (1)). At step 207, the method determines that header(1) has contents, and at step 208 uses the From header contents (sip:Watson@bell-tel.com) as the index into the Request Line table in the MDC 115. At step 210, the method determines that the table contents (see FIG. 4) do not match the contents of the INVITE header in the message. Because the contents did not match, the method uses the To header contents (<sip:simpson@bell-tel.com>) to access the Request Line table in the MDC 115. Because <sip:simpson@bell-tel.com> has not been used as index information into the Invite table before, the Invite table at that index does not match the contents of the INVITE message in the header. At step 214, the method stores the contents of the INVITE message in the Request Line table. At step 216, the method increments "i" and at step 218 determines that there are more headers to be processed.

At step 207, the method determines that header(2) has contents. At step 208, the method uses the contents of the From header as the index information into the Via table of the MDC 115. As shown in FIG. 4, the contents of the Via table are "SIP/2.0/UDP saturn.bell-tel.com." Thus, at step 210, the method determines that the contents of the Via table in the MDC 115 match the contents of the Via

header in the message (header(2)). Because there is a match, there is no need to use the second level of indexing (To header) into the Via table. At step 212, the method determines that header(2) is not the type of message and does not contain the index information and blanks the contents of the Via header (step 222). Next, the method proceeds to step 216 to increment "i At step 218, the method determines that there are more headers to be processed and returns to step 207. The method repeats steps 207, 208, 210, 214, 216 and 218 until all headers have been processed. The method checks the contents of each header in the message against its corresponding table in the MDC 115 using first the 'From" header contents as the index and if there is no match, the "To" header contents are used to again access the table. If there is no match for the second time, the contents of the header in the message are stored in the table at the index corresponding to the To header information. If the message header is a new header (i.e., a header not already used in a message before such as the Subject and Content Type headers in the current example), the contents of the header in the message are stored in the MDC table twice, once at the location corresponding to the From header index information and again at the location corresponding to the To header index information. After each header has been processed, the MDC tables contain the information shown in FIG. 6.

In the current example, the only match found from comparing the header contents in the message to the header contents in the MDC tables is the "Via" header. (Compare FIGs. 4 and 6). The MDC 115 was updated for all other fields, with the combination of the "From" and "To" header contents used for the index information. The compressed INVITE message below is sent to the P-CSCF 102.

INVITE sip:simpson@springfield.bell-tel.com SIP/2.0
Via:

From: T. Watson <sip:watson@bell-tel.com>
To: H. Simpson <sip:simpson@bell-tel.com>
Call-ID: 662606876@saturn.bell-tel.com
CSeq: 6 INVITE
Contact: <sip:watson@saturn.bell-tel.com>
Subject: Mr. Simpson, Going to Moe's?
Content-Type: application/sdp

Upon receipt of the compressed INVITE message, the TCCB layer 128 in the P-CSCF 102 invokes the decompression method. Upon determining that the message is an INVITE message, the method uses the "From" and "To" headers to access the CNC 127. In the current example, the method fills in the contents of the Via header in the message with information in the Via table of the CNC 127. For the other headers in the message, the tables in the CNC 127 are updated with the current information.

Those skilled in the art will recognize that various modifications and variations can be made in the apparatus and method of the present invention and in construction of this apparatus without departing from the scope or spirit of this invention. For example, error checking and handling could be incorporated into the method. The method could perform a checksum on each message before compressing and send the result to the decompressor in the receiving device with the message. Upon decompressing the message, a checksum could be performed on the reconstructed message and the result compared with the original value. If an error is detected, the application layer 110, 124 could be informed using error handling capabilities.

Additionally, bitwise tags may be used for the message headers rather than the text format as described herein. For example, when

applied to SIP, the TCCB layer in both nodes (mobile device 100 and P-CSCF 102) may be initialized with the following mappings:

Via:	0x000158
From:	0x000258
To:	0x000358
Call-ID:	0x000458
Cseq:	0x000558
Contact-length:	0x000658

When sending the compressed message "Via:" would be replaced with 0x000158, and so on. Alternatively, Huffman encoding could be used for the tag format.

As seen in the example of the INVITE message, multiple levels of indexing may be used to cache and retrieve information. Additional information may be used from the message to provide a further level of granularity, where required. The number of levels of indexing implemented will be dependent on the optimum compression rates depending on the characteristics of the protocol. For example, in order to improve the efficiency of the method, a three level indexing could provide further compression

(e.g. From: contents

+To: contents

+Cseq: contents).

Further compression could be achieved in a system where a mandatory header scheme is implemented. A mandatory header scheme is a scheme whereby certain headers must be included in every message. In such a system, further compression could be achieved by not including the header title in the compressed message where the header contents have been blanked.

What Is Claimed Is:

1. In a sending device having a memory, a method of compressing a message having a plurality of headers comprising the steps of:

determining a characteristic of the message;

based on the characteristic of the message, determining which part of the message contains index information into the first memory;

for each of the plurality of headers in the message,

using the index information into the memory to determine whether contents of the header are stored in the memory; and

if contents of the header are stored in the memory, blanking the contents of the header in the message.

2. The method of claim 1 further comprising the step of if the contents of the header are not stored in memory, storing the contents of the header in memory.

3. The method of claim 2 wherein the index information corresponds to contents of at least one predetermined header in the message.

4. The method of claim 1 wherein the contents of the header in the message are not blanked if the header indicates message type or contains the index information.

5. In a receiving device having a memory, a method of decompressing a message having a plurality of headers comprising the steps of:

determining a characteristic of the message;

based on the characteristic of the message, determining which part of the message contains index information into the memory;

for each of the plurality of headers in the message,

determining whether contents of the header is blank; and

if the contents of the header are blank, using the index information into the memory to retrieve information to fill in the contents of the header.

6. The method of claim 5 further comprising the step of if the contents of the header are not blank, using the index information into the memory to store the contents of the header in an appropriate location in the memory.

7. The method of claim 5 wherein the index information corresponds to contents of at least one predetermined header in the message.

8. An apparatus for compressing a text message before transmission over an air interface, the apparatus having a software architecture comprising an application layer, a text based protocol layer, and a physical layer, the apparatus further comprising:

a text based compression/decompression layer for

determining a characteristic of the message;

based on the characteristic of the message, determining which part of the message contains an index information into the first memory;

for each of a plurality of headers in the message,

using the index information into the memory to determine whether contents of the header are stored in the memory; and

if contents of the header are stored in the memory, blanking the contents of the header in the message.

9. An apparatus for decompressing a text message, the apparatus having a software architecture comprising an application layer, a text based protocol layer, and a physical layer, the apparatus further comprising:

a text based compression/decompression layer for

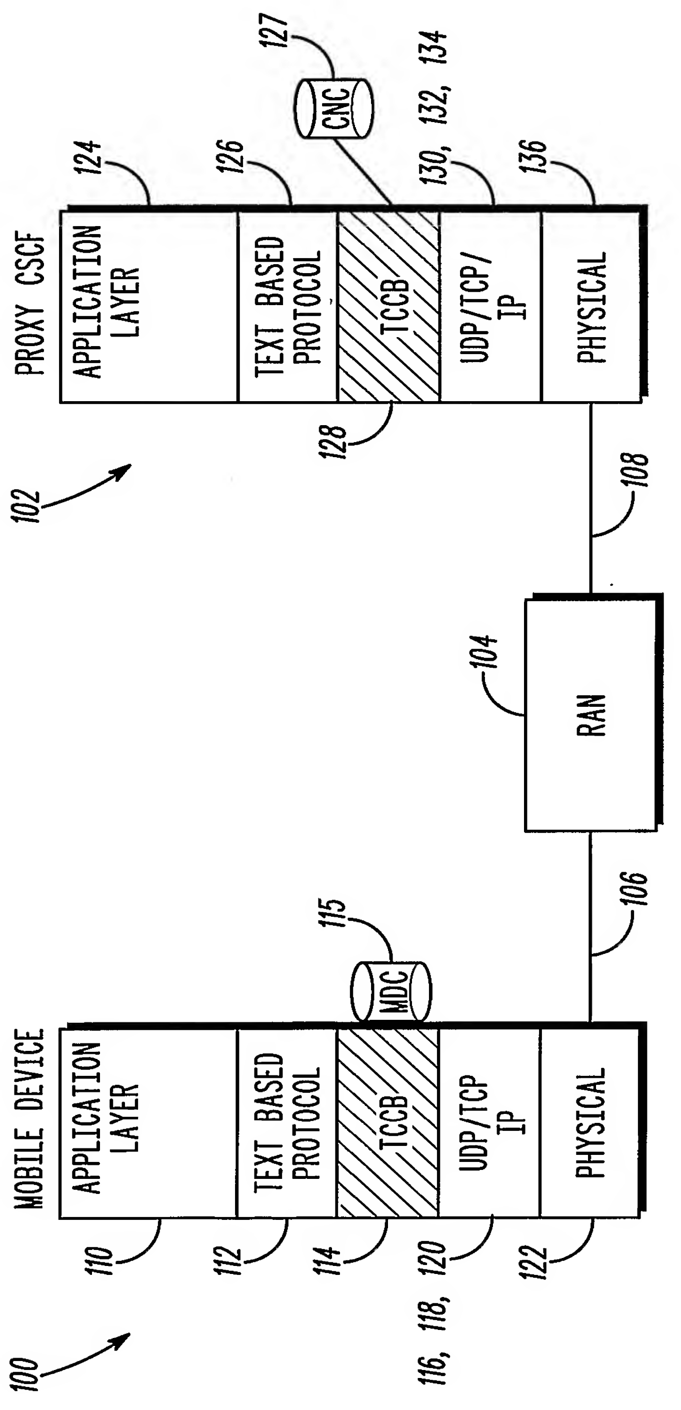
determining a characteristic of the message;

based on the characteristic of the message, determining which part of the message contains an index information into the memory;

for each of a plurality of headers in the message,

determining whether contents of the header are blank; and

if the contents of the header are blank, using the index information into the memory to retrieve information to fill in the contents of the header.



PROXY CSCF
APPLICATION LAYER
TEXT BASED PROTOCOL
TCCB
UDP/TCP/IP
PHYSICAL
CNC
MOBILE DEVICE
APPLICATION LAYER
TEXT BASED PROTOCOL
TCCB
UDP/TCP IP
PHYSICAL
MDC
RAN
100
102
104
106
108
110
112
114
115
116, 118, 120
122
124
126
127
128
130, 132, 134
136

FIG. 1

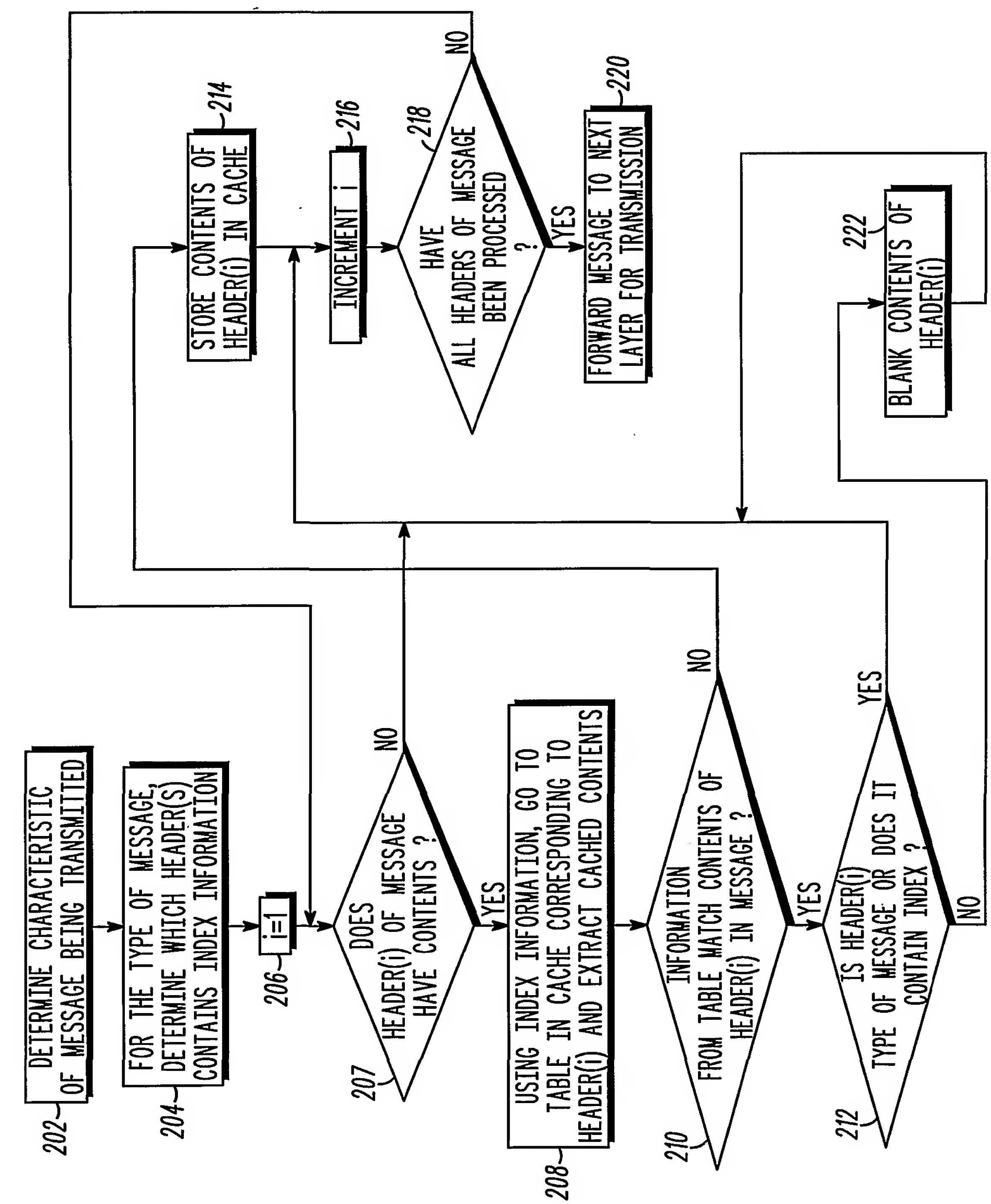

DETERMINE CHARACTERISTIC OF MESSAGE BEING TRANSMITTED
202
FOR THE TYPE OF MESSAGE, DETERMINE WHICH HEADER(S) CONTAINS INDEX INFORMATION
204
i=1
206
DOES HEADER(i) OF MESSAGE HAVE CONTENTS ?
207
NO
YES
USING INDEX INFORMATION, GO TO TABLE IN CACHE CORRESPONDING TO HEADER(i) AND EXTRACT CACHED CONTENTS
208
INFORMATION FROM TABLE MATCH CONTENTS OF HEADER(i) IN MESSAGE ?
210
NO
YES
IS HEADER(i) TYPE OF MESSAGE OR DOES IT CONTAIN INDEX ?
212
YES
NO
STORE CONTENTS OF HEADER(i) IN CACHE
214
INCREMENT i
216
HAVE ALL HEADERS OF MESSAGE BEEN PROCESSED ?
218
NO
YES
FORWARD MESSAGE TO NEXT LAYER FOR TRANSMISSION
220
BLANK CONTENTS OF HEADER(i)
222

FIG. 2

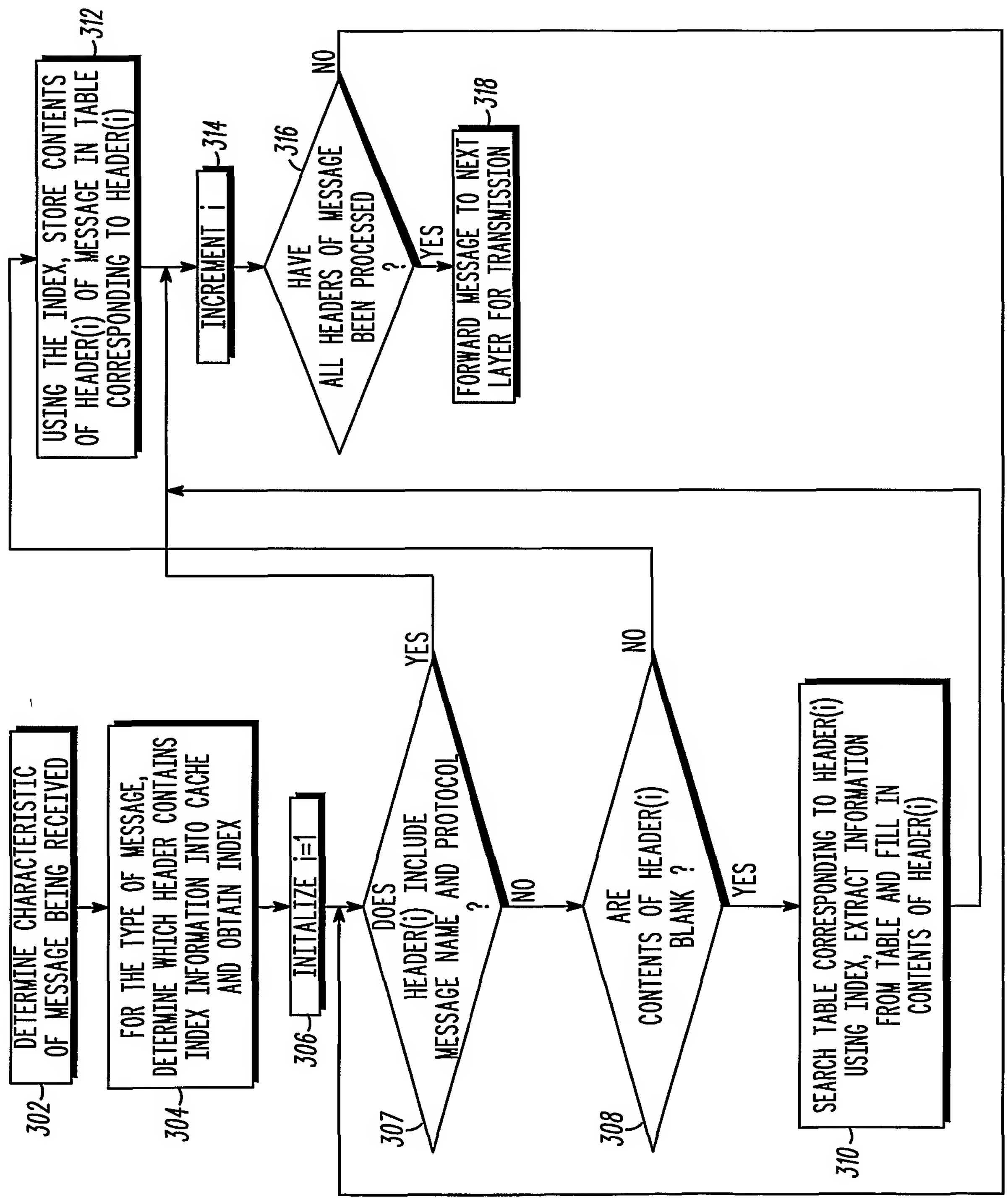

DETERMINE CHARACTERISTIC OF MESSAGE BEING RECEIVED
302
FOR THE TYPE OF MESSAGE, DETERMINE WHICH HEADER CONTAINS INDEX INFORMATION INTO CACHE AND OBTAIN INDEX
304
INITALIZE i=1
306
DOES HEADER(i) INCLUDE MESSAGE NAME AND PROTOCOL ?
307
YES
NO
ARE CONTENTS OF HEADER(i) BLANK ?
308
NO
YES
SEARCH TABLE CORRESPONDING TO HEADER(i) USING INDEX, EXTRACT INFORMATION FROM TABLE AND FILL IN CONTENTS OF HEADER(i)
310
USING THE INDEX, STORE CONTENTS OF HEADER(i) OF MESSAGE IN TABLE CORRESPONDING TO HEADER(i)
312
INCREMENT i
314
HAVE ALL HEADERS OF MESSAGE BEEN PROCESSED ?
316
NO
YES
FORWARD MESSAGE TO NEXT LAYER FOR TRANSMISSION
318

FIG. 3

TABLE	INDEX	CONTENTS
REQUEST LINE	sip:watson@bell-tel.com	sip:bell-tel.com SIP/2.0
VIA	sip:watson@bell-tel.com	SIP/2.0/UDP saturm.bell-tel.com
FROM	sip:watson@bell-tel.com	<sip:watson@bell-tel.com> TAG=19a1
CALL ID	sip:watson@bell-tel.com	70710@saturn.bell-tel.com
CSEQ	sip:watson@bell-tel.com	1 REGISTER
CONTACT	sip:watson@bell-tel.com	<sip:watson@saturn.bell-tel.com:3890; transport=udp>
EXPIRES	sip:watson@bell-tel.com	7200

FIG. 4

TABLE	INDEX	CONTENTS
REQUEST LINE	sip:watson@bell-tel.com	sip:bell-tel.com SIP/2.0
VIA	sip:watson@bell-tel.com	SIP/2.0/UDP saturn.bell-tel.com
FROM	sip:watson@bell-tel.com	<sip:watson@bell-tel.com> TAG=19a1
CALL ID	sip:watson@bell-tel.com	70710@saturn.bell-tel.com
CSEQ	sip:watson@bell-tel.com	1 REGISTER
CONTACT	sip:watson@bell-tel.com	<sip:watson@saturn.bell-tel.com:3890; transport=udp>
EXPIRES	sip:watson@bell-tel.com	7200

FIG. 5

TABLE	INDEX	CONTENTS
REQUEST LINE	sip:watson@bell-tel.com +sip:simpson@bell-tel.com	Sip:bell-tel.com SIP/2.0 sip:simpson@springfield.bell-tel.com SIP/2.0
VIA	sip:watson@bell-tel.com	SIP/2.0/UDP saturn.bell-tel.com
CALL-ID	sip:watson@bell-tel.com +sip:simpson@bell-tel.com	70710@saturn.bell-tel.com 662606876@saturn.bell-tel.com
CSEQ	sip:watson@bell-tel.com +sip:simpson@bell-tel.com	1 REGISTER 6 INVITE
CONTACT	sip:watson@bell-tel.com +sip:simpson@bell-tel.com	<sip:watson@saturn.bell-tel.com:3890; transport=udp> <sip:watson@saturn.bell-tel.com>
SUBJECT	sip:watson@bell-tel.com +sip:simpson@bell-tel.com	MR. SIMPSON, GOING TO MOE'S? MR. SIMPSON, GOING TO MOE'S?
CONTENT TYPE	sip:watson@bell-tel.com +sip:simpson@bell-tel.com	APPLICATION/sdp APPLICATION/sdp

FIG. 6

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 02/20545

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, INSPEC, COMPENDEX, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CASNER S ET AL: "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links" IETF, 27 July 1998 (1998-07-27), XP002125101 * pag. 4, par. 3.1 "The basic idea" * * pag. 6, par. 3.3 "The protocol" * * pag. 8 par. 3.3.1 "FULL HEADER" * * pag. 10 par. 3.3.2 "COMPRESSED RTP" * * pag. 14 par. 3.3.3 "COMPRESSED UDP" * --- -/--	1-9

[X] Further documents are listed in the continuation of box C.

[ ] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
3 September 2002	16/09/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Falò, L

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	STORZ W ET AL: "Transmitting time-critical data over heterogeneous subnetworks using standardized protocols" COMPUTER COMMUNICATIONS AND NETWORKS, 1995. PROCEEDINGS., FOURTH INTERNATIONAL CONFERENCE ON LAS VEGAS, NV, USA 20-23 SEPT. 1995, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 20 September 1995 (1995-09-20), pages 82-87, XP010200309 ISBN: 0-8186-7180-7 * pag. 82, par. 1 "Introduction" * * pag. 82, par. 2 "Basic concept" * * pag. 83 par. 3 "Header compression" * * pag. 83 par. 3.1 "Method and parameters of compression" * ---	1-9
X	DEGERMARK M ET AL: "LOW-LOSS TCP/IP HEADER COMPRESSION FOR WIRELESS NETWORKS" WIRELESS NETWORKS, ACM, US, vol. 3, no. 5, 1 October 1997 (1997-10-01), pages 375-387, XP000728935 ISSN: 1022-0038 * pag. 376 par. 2 "Header Compression" * ---	1-9
A	PERKINS S J ET AL: "Dependency removal for transport protocol header compression over noisy channels" COMMUNICATIONS, 1997. ICC '97 MONTREAL, TOWARDS THE KNOWLEDGE MILLENNIUM. 1997 IEEE INTERNATIONAL CONFERENCE ON MONTREAL, QUE., CANADA 8-12 JUNE 1997, NEW YORK, NY, USA,IEEE, US, 8 June 1997 (1997-06-08), pages 1025-1029, XP010227258 ISBN: 0-7803-3925-8 * pag. 1026 par. 3.1 "The VJ Algorithm" * * pag. 1026 par. 3.2 "Sync. Loss" * * pag. 1027 par. 4.1 "VJ Connection ID Compression" * * pag. 1028 par. 4.2 "Interpacket Dependency removal" * ---	1-9
A	"SESSION HEADER COMPRESSION SCHEME USING DIFFERENTIAL RUN-LENGTH ENCODING" IBM TECHNICAL DISCLOSURE BULLETIN, IBM CORP. NEW YORK, US, vol. 40, no. 12, 1 December 1997 (1997-12-01), pages 125-126, XP000754111 ISSN: 0018-8689 the whole document --- -/--	1-9

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	JACOBSON V: "COMPRESSING TCP/IP HEADERS FOR LOW-SPEED SERIAL LINKS" NETWORK WORKING GROUP REQUEST FOR COMMENTS, XX, XX, February 1990 (1990-02), XP002901830 * par. 3 "The compression algorithm" pages 4-13 *	1-9
